INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION

INTERNATIONAL CENTRE FOR SCIENCE AND HIGH TECHNOLOGY
c/o INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS 34100 TRIESTE (ITALY) VIA GRIGNANO, 9 (ADRIATICO PALACE) P.O. BOX 586 TELEPHONE 040 224572 TELEFAX 040 224573 TELEX 460449 APH I

SMR/474 - 21

**COLLEGE ON**
**"THE DESIGN OF REAL-TIME CONTROL SYSTEMS"**
**1 - 26 October**

# *EXTENSIONS TO OS-9*
## *(Part II)*

**H. VON DER SCHMITT**
**C.E.R.N.**
**PPE Division**
**CH-1211 Geneva 23**
**Switzerland**

## Interprocess communication

purposes: data exchange
synchronisation
mutual exclusion

- OS-9 signals

F$Send — to send a signal to a specified process.

F$Sleep — to await a signal.

F$Icpt — to install a routine for handling special signals.

Standard signal codes:

| | |
|---|---|
| kill | |
| wakeup | used by driver's IRQ routine |
| interrupt | control-C on terminal |
| abort | control-E on terminal |



- Signals carry a 16-bit information (signal code; 6809: 8-bit).
- They are process specific, rather than system global (exception: ID = 0).
- They are queued to the receiving process (were not in 6809).
  Exception: wakeup lost if process running!
- Processes can mask signals (not in 6809). Caution with real time response!

Signals are primarily useful for the internal working of I/O system. Careful when using explicitely for synchronisatio

- OS-9 events

system-global 32-bit counters,
use as semaphores on shared resources
(e.g. data modules), and for signaling.

F$Event system call provides
Signal primitive
Wait primitive
Maintainance functions (create etc.)

Ev$Creat has arguments:
event name ,e.g. "data_ready"
initial value ,e.g. ∅
signal increment ,e.g. 1
wait increment ,e.g. -1

Ev$Link has argument:
event name ,e.g. "data_ready"

- both return: event id for name.



Ev$Signl has arguments:
event id (from create or link)

- it increments the counter by the amount given in create

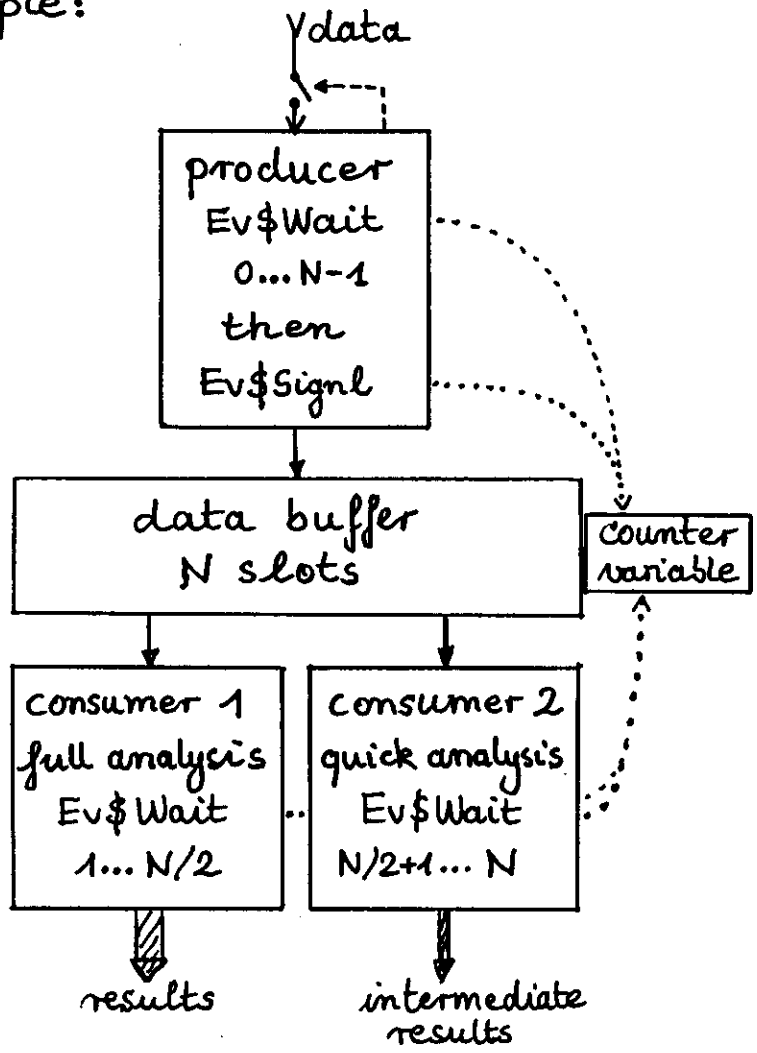
Ev$Wait has arguments:
event id (from create/link)
minimum value ,e.g. 1
maximum value ,e.g. ∞

indivisible operation:
- if counter not in range min...max, suspends process until in range (then re-activates)
once in range, decrements the counter by amount given in create.

- returns event counter value for checks (event-wait could also have terminated by a signal from F$Send).

(There are more – don't discuss here.)

Example:




Events usually preferred to signals:

system global, faster, more flexible.



- Named pipes

embedded in the I/O system

system-global → even network-global with OSI Net

for message exchange, includes message buffering and synchronisation.

multiple readers and writers per pipe possible.

device name: /pipe

one-level directory: file e.g. /pipe/data

organised as a fifo in memory by pipeman.

I$Create, I$Open } to create/open

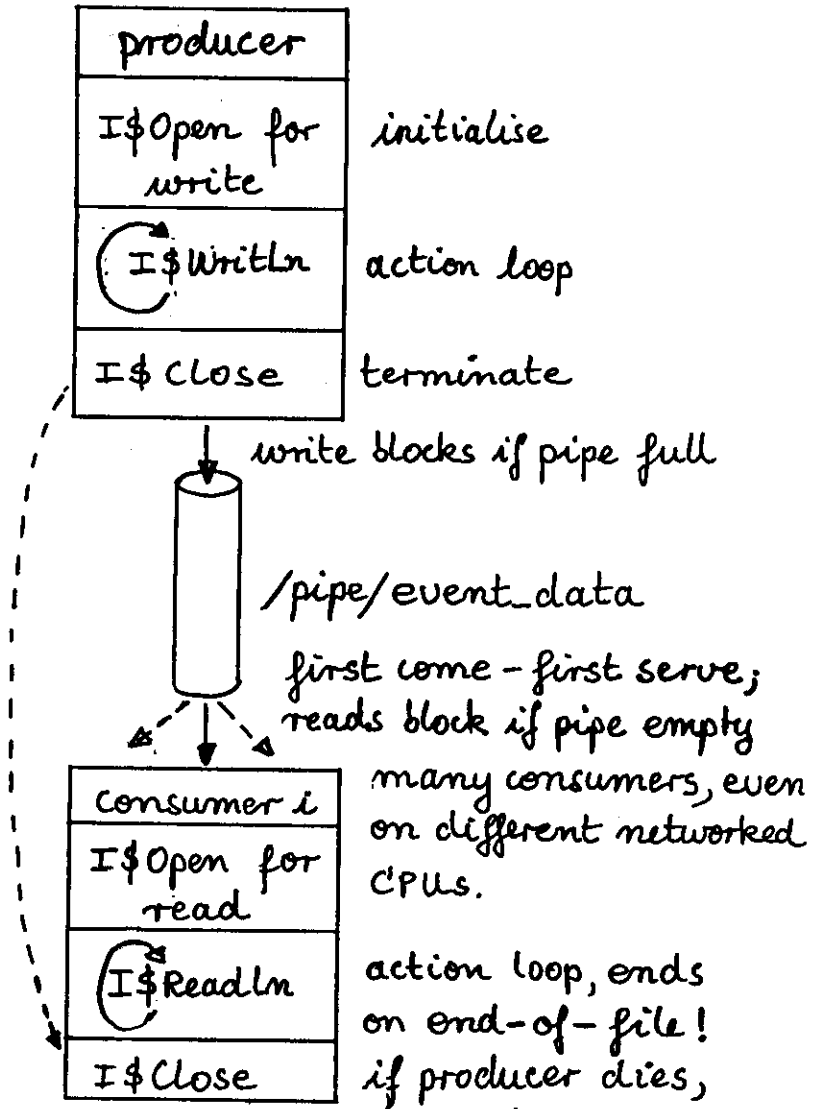
I$WritLn for writing – blocks if pipe full

I$ReadLn for reading – blocks if pipe empt

I$GetStt(SS_Ready) asynchronous signal if data available

all as for other files or terminal!

Example of named pipes:






- Data modules

Easy means to have named module for use as data buffer, set of constants, etc.

F$DatMod creates it. Parameter: size.

F$Link links to it, as for other types of modules.

If constants, one can use load to get it from disk to memory, as for code.
Protected with CRC, as all modules.
Use fixmod for CRC update if not constants.

## Networking

A means to connect several computers over some distance (≳ 10m ... worldwide). Generally over serial line, using some protocol sending/receiving packets of data.
(Why not use the processor bus? → distance too long, too processor specific!)

OS-9 offers two protocols by means of extra file managers/drivers. Often Ethernet is used: serial, 10 Mbit/sec, packets ≤ 1500 byte

1) "TCP/IP" protocols: a standard that runs on all bigger computers, all UNIX systems e.g., and also OS-9. Allows e.g. file transfer and remote logon between <u>any</u> two systems, whether of same typ or not.
File manager IPMAN + much more; ~300 kbytes.

2) OSGNet: confined to OS-9 systems. (4)
Very well integrated into OS-9: allows remote execution of any I/O system call I$....
The network itself is a device, supported by file manager NFM, drivers, descriptors.
Device e.g.: <u>/enet</u>
The "nodes" (computers) along the network appear like a first-level file directory.
Path to node e.g.: <u>/enet/cpu5</u>
On the remote node, one can then access any file by appending its local path name:
Complete path e.g.:

| | |
|---|---|
| /enet/cpu5/p | remote printer |
| /enet/cpu5/d0/cmds/x | remote file |

even:

| | |
|---|---|
| /enet/cpu5/xnet/cpu0/... | i.e. built-in "bridge". |

There is a utility for remote login:

```
chp /enet/cpu5
```

One can use remote pathnames in all sorts of commands, e.g.

```
chd /enet/cpu5/dØ/help
load /enet/cpu5/rØ/cmds/edit
list xx >/enet/cpu5/p
```

The last example shows how easy remote interprocess communication is with pipes and OS9 Net:

```
analyse </enet/cpu5/pipe/event_data
```

The process, analyse, is completely unaware of where its data come from - all done by rerouting on the shell level. Could as well come from a file:

```
analyse </dØ/testdata .
```







How OS9 Net works

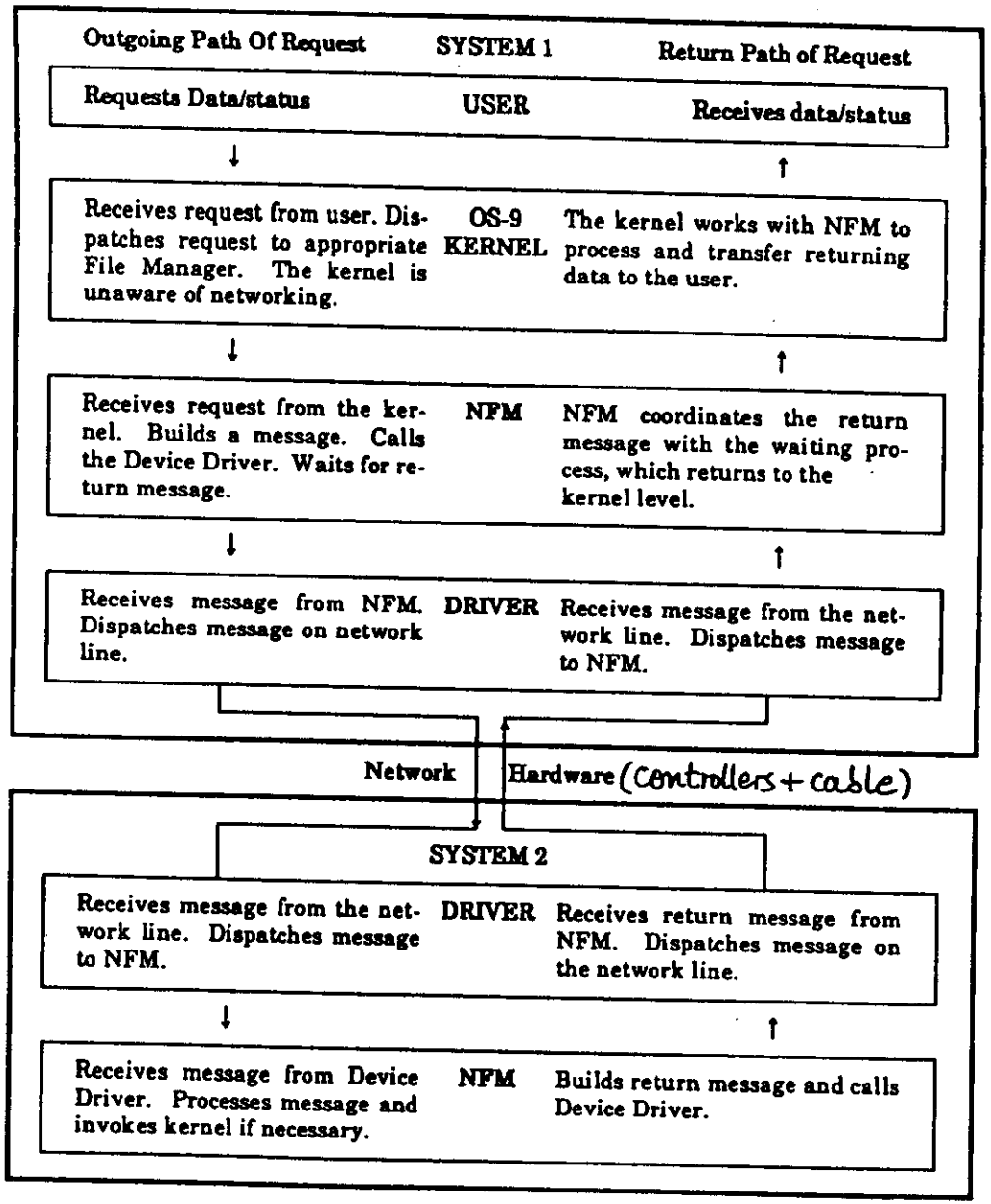



Figure 26: Tracing a user request through the network

## Enhancements to the shell

OS-9/68K

- understands the usual wildcard characters: * for any string
  ? for any one character

  e.g. list *.c — any file with .c at end
  list prog* — any file starting with prog
  dir ??? — any file with 3 characters

- provides environment variables, which can be retrieved by processes.
  Utility setenv allows to define the variables.
  e.g.
  setenv PATH /dd/cmds:/h0/objs:/xx
  defines variable PATH, used during load as a sequence of paths to be searched.
  : acts as separator in sequence list.

## OS-9000 (on 68k, 80386, 88000)



- shell remembers the commands typed in earlier. Can recall them with cursor keys, and edit them if necessary.
- shell command scripts can use parameters
- IOMAN handles logical path names defined with utility alias, eg.
  alias /h0/mine /dd,
  then valid inside this shell for everything
- faster disk I/O by cache in memory.

OS-9000 is written in C, "almost entirely", thus portable to new processor types (like UNIX is); but not by us - needs a C compiler first - and real specialists ...

.—•—.